微处理机自学读本

〔美〕希思公司进修教材

清华大学自动化系电子学教研组
北京无线电技术研究所 合译

人民邮电出版社

微处理机自学读本

〔美〕 布思公司进修教材

清华大学自动化系电子学教研组
北京无线电技术研究所
合译

000059

人民邮电出版社

HEATHKIT CONTINUINC EDUCATION

Individual Learning Program

MICROPROCESSORS

HEATH COMPANY 1977

内 容 提 要

本书是美国希思（Heath）公司出版的进修教材—《微处理机自学读本》，主要是为学习6800微处理机而编写的。可供使用微处理机的工人、技术人员和研究人员阅读，也可供初学者作为自学入门参考书。全书共分十个单元：第一～六单元为微处理机的基本原理部分，介绍得比较系统，内容深入浅出，重点突出，概念清楚；第七～八单元为接口部分，主要讲解MPU与其他电路的接口，并给出足够的通用资料熟悉接口实验；第九～十单元是作者精心编制的软、硬件及系统应用实验，配合前面八个单元的理论学习，可以获得很好的学习效果；最后还附有指令系统和ET-3400学习器的原理图。

微处理机自学读本

〔美〕希思公司进修教材

清华大学自动化系电子学教研组
北京无线电技术研究所 合译

人民邮电出版社出版

北京东长安街27号

天津新华印刷一厂印刷

新华书店北京发行所发行

各地新华书店经售

开本：787×1092 1/16　1983年 2月 第一版

印张：27 12/16页数：222 1984年7月天津第2次印刷

字数：691 千字　插页：1　印数：15,001— 70,000册

统一书号：15045·总2644—有5269

定价： 2.60 元

译 者 说 明

微处理机自七十年代初期问世以来，在不到十年的时间，得到了飞快的发展。其应用范围已扩展到了各行各业——从航天技术到日常生活，从工业上的应用到儿童玩具，都有微处理机的踪影。据有关人士估计，这种器件不仅将推动数字电子领域的变革，而且对人们的生活方式也将带来深刻的影响。

微处理机所以能获得广泛应用，除了价格低廉，可靠性高之外，还有体积小、功能强、灵活性大等优点。与微处理机配套的芯片种类越来越齐全和软件的日益丰富也是促进其广泛应用的重要因素。

微处理机的种类很多，目前比较流行的是：Motorola公司的M6800系列，Intel 公司的8080系列和Zilog公司的Z-80系列，这些都是 8 位字长的机器，也有单片机及12位的微型机。从近年的生产情况看来，这三家公司都生产出了16位的微处理机，即Motorola的M68000系列，Zilog的Z8000系列，以及较早一些的Intel8086，有取代小型机的趋势。当然，利用大规模集成电路小型机也在微型化，如与PDP-11相同的LSI-11，与NOVA机功能相同的μN-OVA等。从应用方面陈列机和多机系统是正在研究的课题，但大量生产和应用的仍然为8位机。

本书主要是为学习M6800微处理机而编写的，它是一本较好的微处理机教材，可供使用微处理机的工人、技术人员、研究人员阅读，作为自学入门参考书。

本书的第一～六单元为微处理机的基本原理部分；第七～八单元为接口部分；第九～十单元为软、硬件及系统应用的实验；最后还附有指令系统及数据图表（略）的资料，内容较为完整。

本书原理部分讲得比较系统，内容深入浅出，重点突出，简明扼要。书中还附有大量插图，十分形象。对于有一些数字电路知识的人都能看懂，从中可较快地学到微处理机的基本原理。

第九、十两单元是作者精心编制的实验，读者如有一台与书配套的学习器，则通过实验操作就能较快、较好地掌握软件知识及应用知识。为了读者学习方便。在第九单元后面附加了ET-3400学习器的键盘使用说明。但如果没有学习器，读者也可自学这两单元，可将第九单元作为软件习题及题解来学习，将第十单元作为应用资料来学习。

本书内容偏重于基础。所以在学完之后尚要继续了解其他外围芯片及程序设计知识，以适应具体应用。接口片子书中只讲了PIA，其他ACIA及Modem 等没有提到，读者以后可进一步学习。

本书虽主要是为M6800系列所写，但由于该书原理讲的较为系统、扎实，所以学好这本书对学习其它类型的微处理机也甚为有益，能收到触类旁通的效果。

与本书配套的资料尚有：ET-3400学习器、用户手册、视听教材（有画片及录音带）、指令卡和附录B（数据表），需要时，读者可向有M6800微处理机设备的单位联系。

本书由清华大学自动化系电子学教研组及北京无线电技术研究所联合翻译，由于时间及水平所限，错误在所难免，恳请批评指正。

译者

本书任务：

当读者学习本书后，将能：

1.编制典型微处理机的程序。

2.完成典型微处理机与外界的接口。

学 习 提 纲

1

数 制 和 编 码

一、引　　言

二进制数及编码是所有微处理机的基本语言，八进制数、十六进制数用二进制数表示和处理极为方便。因此，建立这些数和码的雄厚基础，对于深入学习微处理机是非常重要的。

这里，首先重温一下十进制数、然后再将数的基本概念引伸到二进制、八进制、十六进制数。充分地理解它们将有助于掌握微处理机的许多数字编码。虽然，本单元只介绍了一些有关数码的基本知识，但当你通过下几个单元学习之后，将会更加熟悉它们。

下一节列有本单元的学习任务，以使你了解其学习内容。请务必按本单元指出的学习步骤逐项完成，希望你每完成一步，就在学习步骤中画的空格（□）上打个√，并记下完成每步所花费的时间。

二、本单元任务

当读者学完本单元之后，将具有下列的知识和本领：

1、给定任意一个十进制数，能够把它转换成相应的二进制、八进制、十六进制和二——十进制（BCD）数。

2、给定任意一个二进制数，能够把它转换成相应的十进制、八进制、十六进制和二——十进制数。

3、给定任意一个八进制数，能够把它转换成相应的十进制和二进制数。

4、给定任意一个十六进制数，能够把它转换成相应的十进制和二进制数。

5、给定任意一个二——十进制数，能够把它转换成相应的十进制和二进制数。

6、给一个普通的数码表，能够读出并识别出哪些是二进制码、二——十进制码、格雷码（Gray）、美国信息交换标准代码（ASCII码）和博多码（BAUDOT）。

7、能够把字母或数字转化成ASCII二进制码。反之，也能够把ASCII二进制码转换成相应的字母或数字。

8、能够给下列术语下定义：

底数（Radix）
整数（Integer）
十进制（Decimal）
二进制（Binary）
八进制（Octal）
十六进制（Hexadecimal）
位（比特，Bit）

奇偶性（Parity）
二——十进制（BCD）
格雷码（Gray）
美国信息交换标准代码（ASCII码）
博多码（BAUDOT）
最高位（MSB）
最低位（LSB）
最大有效数字（MSD）
最小有效数字（LSD）

三、本单元学习步骤

	完成时间
□阅读十进制数制一节	——
□完成复习题（1）～（6）	——
□阅读二进制数制一节	——
□完成复习题（7）～（13）	——
□阅读八进制数制一节	——
□完成复习题（14）～（19）	——
□阅读十六进制数制一节	——
□完成复习题（20）～（25）	——
□阅读二进制编码一节	——
□完成复习题（26）～（36）	——
□完成实验1和2	——
□完成本单元考查	——
□核对考查答案	——

四、十进制数制

通常人们所熟悉的数制是十进制，大约在公元400年左右，由印度数学家首先发明，大约在公元800年，阿拉伯人开始使用它，所以称它为阿拉伯数制。以后，大约在公元1200年传到欧洲，才被命名为“十进制数制”

区别一种数制的基本特征是底数或基数。底数表示所用的字符或数码的数目，这些字符和数码表示数制中量的大小。十进制数制因用0～9十个数码表示量的大小，故底数为10。为了说明所用的数制，应在数的下标注明底数。如4603_{10}表示以10为底数的数制。

1、按位计数法

十进制是有位、数的数制。即：一个数中的每位数都有其特定的权，此权决定其数值的大小，每个位权由底数的n次幂确定，在十进制数制中，位权是10^0（个位）、10^1(十位)、10^2（百位）等等。图1-1表示以10为底的n次幂简表

$10^0 = 1$

$10^1 = 10$

$10^2 = 100$

$10^3 = 1000$

$10^4 = 10,000$

$10^5 = 100,000$

$10^6 = 1,000,000$

$10^7 = 10,000,000$

$10^8 = 100,000,000,$

$10^9 = 1,000,000,000$

图 1-1 以10为底的n次幂的简表

＊ 任何数的 0 次幂均等于 1。

我们设一个数，按照位权来计算其总值。例如，十进制数4603按位记数表示为：$(4\times10^3)+(6\times10^2)+(0\times10^1)+(3\times10^0)=(4\times1000)+(6\times100)+(0\times10)+(3\times1)$

$$=4000+600+0+3$$
$$=4603_{10}$$

也就是说每位数字乘以它所在位置的权，相加则得到所求的数值。

2、**小数（分数）**

到目前为止，我们只讨论了整数部分。整数包括自然数、负数和零。如：0、1、4、7等等。因此，整数只表示一个完整的数。但是，人们经常还要用小数来表示数的大小。

十进制小数也是具有位权的数。它们的权都是10的负次幂。例如，$10^{-1}=\frac{1}{10}=0.1$，$10^{-2}=\frac{1}{100}=0.01$等等。

图 1—2 为10的负 n 次幂的简表(十进制的小数)。

$$10^{-1}=\frac{1}{10}=0.1$$

$$10^{-2}=\frac{1}{100}=0.01$$

$$10^{-3}=\frac{1}{1,000}=0.001$$

$$10^{-4}=\frac{1}{10,000}=0.0001$$

$$10^{-5}=\frac{1}{100,000}=0.00001$$

$$10^{-6}=\frac{1}{1,000,000,}=0.000001$$

图 1-2 10的负n次幂的简表

小数点把一个数分为整数和小数两部分。小数点左边表示整数，它所在位置的权是10^0、10^1、10^2……。小数点右边表示小数部分，小数所在位置的权是10^{-1}、10^{-2}、10^{-3}。如十进制数278.94，用按位计数法表示为：

$$(2\times10^2)+(7\times10^1)+(8\times10^0)+(9\times10^{-1})+(4\times10^{-2})$$

$$=(2\times100)+(7\times10)+(8\times1)+\left(9\times\frac{1}{10}\right)+\left(4\times\frac{1}{100}\right)$$

$$=200+70+8+0.9+0.04=278.94$$

在这个例子中，最左边的数字（2×10^2）是最大有效数字（MSD），因为在确定数值时，它的权最大。最右边数字，称作最小有效数字（LSD），因为在确定数值时，它的权最小。也就意味着当MSD的数字发生变化时，对数值影响最大，而LSD的变化对整个数的影响最小。

3、复习题

（1）在一个数制中，用____表示字符或数码的数。

（2）在十进制数制中，底数或基数是____。

（3）用按位计数法写出下列各数

A、4563_{10}

B、26.32_{10}

C、536.9_{10}

（4）在十进制数制中，小数点被称作____

（5）把下列按位计数法表示的数转换成简化的十进制形式。

A、$(5\times10^1)+(2\times10^0)+(3\times10^{-1})+(8\times10^{-2})$

B、$(4\times10^{-1})+(6\times10^{-2})+(2\times10^{-3})$

C、$(3\times10^3)+(7\times10^2)+(1\times10^1)+(0\times10^0)$

（6）小数点把数的____和____分开。

4、答案：

（1）底数或基数。

（2）10。

（3）A、$4563_{10}=4000+500+60+3$

$=(4\times10^3)+(5\times10^2)+(6\times10^1)+(3\times10^0)$

B、$(2\times10^1)+(6\times10^0)+(3\times10^{-1})+(2\times10^{-2})$

C、$(5\times10^2)+(3\times10^1)+(6\times10^0)+(9\times10^{-1})$

（4）十进制小数点

（5）A、$(5\times10^1)+(2\times10^0)+(3\times10^{-1})+(8\times10^{-2})=(5\times10)+$

$+(2\times1)+(3\times1/10)+(8\times1/100)=50+2+0.3+0.08$

$=52.38_{10}$

B、0.462_{10}

C、3710_{10}

（6）整数、小数。

五、二进制数制

按位计数法中最简单的是二进制。它只包括两个元素或状态，即1和0。二进制中的0

和 1 与十进制中的 0 和 1 具有一样的数值。

由于二进制数很简单，微处理机就用它来处理数据。人们用称作位（bit）的二进制码来表示二进制数据（bit是英语binary digit的缩写）。微处理机对称作字的一组二进制码进行运算。比如，二进制数11101101即为八位。

1、按位计数法

和十进制数制一样，二进制数的每一位所在的位置均带在一个确定数值大小的特定权。人们用数制底数的 n 次幂来表示每位的权。为了计算总数值，必须考虑具体的二进制码和它所在位置的权。图 1 — 3 是 2 的 n 次幂表。例如，二进制数110101按位计数则为：

$$(1\times2^5)+(1\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$$

要确定二进制数110101所表示的十进制数，就要把二进制数的每一位乘以这一位的位权，而后，相加则得：

$$(1\times32)+(1\times16)+(0\times8)+(1\times4)+(0\times2)+(1\times1)$$
$$=32+16+0+4+0+1=53_{10}$$

$2^0=1_{10}$	$2^6=64_{10}$
$2^1=2_{10}$	$2^7=128_{10}$
$2^2=4_{10}$	$2^8=256_{10}$
$2^3=8_{10}$	$2^9=512_{10}$
$2^4=16_{10}$	$2^{10}=1024_{10}$
$2^5=32_{10}$	$2^{11}=2048_{10}$

图 1-3　2 的n次幂的简表

人们把二进制小数表示为 2 的负 n 次幂。图 1 — 4 为 2 的负 n 次幂的简表。二进制数 0.1101按位表示为：

$$(1\times2^{-1})+(1\times2^{-2})+(0\times2^{-3})+(1\times2^{-4})$$

将二进制数0.1101转化成十进制数，只要把二进制的每一位乘以该位所在的位权，然后相加则得：

$$\left(1\times\frac{1}{2}\right)+\left(1\times\frac{1}{4}\right)+\left(0\times\frac{1}{8}\right)+\left(1\times\frac{1}{16}\right)$$
$$=0.5+0.25+0+0.0625=0.8125_{10}$$

在二进制中，小数点称作二进制小数点。

$2^{-1}=\frac{1}{2}=0.5_{10}$	$2^{-5}=\frac{1}{32}=0.03125_{10}$
$2^{-2}=\frac{1}{4}=0.25_{10}$	$2^{-6}=\frac{1}{64}=0.015625_{10}$
$2^{-3}=\frac{1}{8}=0.125_{10}$	$2^{-7}=\frac{1}{128}=0.0078125_{10}$
$2^{-4}=\frac{1}{16}=0.0625_{10}$	$2^{-8}=\frac{1}{256}=0.00390625_{10}$

图 1-4　2 的负n次幂简表

2、二进制数和十进制数的转换

在使用微处理机的过程中，经常需要计算二进制数所表示的十进制数值，或把一个十进制数转化成相应的二进制数。下面介绍如何实现这种转换。

（1）二→十进制转换

把二进制数转换成相应的十进制数，只要将二进制中出现 1 的所在位权相加即可。整数和小数位权如下：

整数

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

小数

2^{-1}	2^{-2}	2^{-3}
.5	.25	.125

二进制小数点

例如，把二进制数1010转换成相应的十进制数。因为1010没有小数点出现，所以是整数，也就是说，小数点在该数的右边。最右边的一位称最低位（LSB），它的位权最小为 $2^0=1$。最左边的一位称最高位（MSB），因为在确定数值时，它表示的位权最大。在这个例子中，它的位权是 $2^3=8$。要得到总的数值，需把二进制出现 1 的位权值相加即 2^3 与 2^1相加，得到的十进制数是10。

二进制数	1	0	1	0	
位权值	2^3	2^2	2^1	2^0	
十进制数	8	+ 0	+ 2	+ 0	$=10_{10}$

为了进一步说明这个转换过程，我们再举一例，把二进制数101101、11转换成相应的十进数：

二进制数	1	0	1	1	0	1	1	1	
位权值	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	
十进制数	32	+ 0	+ 8	+ 4	+ 0	+ 1	+ 0.5	+ 0.25	= 45.75

（2）十→二进制转换

把一个十进制的整数依次除以所需要的底数，就能够转换成不同底数的数。比如：为了把十进制的数转换成相应的二进制数，只要把十进制数依次除以 2 并记下每次所得的余数（余数总是 1 或 0）所得的余数即为相应的二进制数。

例如，把十进制数25转换成二进制数：

$25\div2=12$	余数	1←LSB
$12\div2=6$		0
$6\div2=3$		0
$3\div2=1$		1
$1\div2=0$		1←MSB

把十进制数除以 2 并记下余数，所得的商再除以 2，并记下余数，然后再把所得的商除以 2，并记下余数，如此不断继续下去，直到商得 0 为止。然后，收集余数，以余数的末位（MSB）开始，直到余数的第一位（LSB）为止，所以$11001_2=25_{10}$。注意：余数一定要按颠倒顺序收集，即：第一位余数是最低位，而末位余数是最高位。

为了进一步说明这个问题。，再举一例，将十进制数175转换成相应的二进制数。

$175\div2=87$　余数　1←LSB

$87 \div 2 = 43$	1
$43 \div 2 = 21$	1
$21 \div 2 = 10$	1
$10 \div 2 = 5$	0
$5 \div 2 = 2$	1
$2 \div 2 = 1$	0
$1 \div 2 = 0$	1 ←MSB

不断除下去，直到结果得 0 为止。收集余数，得到$10101111_2 = 175_{10}$。

要将一个十进制小数转换成不同底数或基数的数时，则应把所需的底数或基数连续不断地乘以该十进制小数，并且记录所得的溢出数（即整数部分）。例如，将十进制数0.3125转换成相应的二进制数，则用 2 重复乘：

$0.3125\times 2 = 0.625 = 0.625$	溢出 0 ←MSB
$0.6250\times 2 = 1.250 = 0.250$	1
$0.2500\times 2 = 0.500 = 0.500$	0
$0.5000\times 2 = 1.000 = 0$	1 ←LSB

这样乘的结果在个位上（即：小数点左边），将得到的 0 或 1，记录下来，就组成相应的二进制小数。这些个位数称做"溢出"。所以，当0.3125乘以 2 时，溢出为 0，这个 0 就是相应的二进制小数的最高位，然后0.625再乘以 2，得到1.25，溢出为 1，当记录数值时，乘积必须减去 1，而在下一步乘法过程中，仅是0.25乘以 2。用同样方法继续下去，直到小数得 0 为止。但有时结果永不为 0，只要转换到所要求的精度为止即可。将起始溢出位写在二进制小数点以后的第一位，（即小数部分的最高位），并继续写到最低位，从最高位到最低位与产生溢出的顺序是一致的。数$0.0101_2 = 0.3125_{10}$。

为了进一步说明该转换过程，再举一例，将十进制小数0.90625转换成相应的二进制数。

$0.90625\times 2 = 1.8125 = 0.8125$	溢出 1 ←MSB
$0.81250\times 2 = 1.6250 = 0.6250$	1
$0.62500\times 2 = 1.2500 = 0.2500$	1
$0.25000\times 2 = 0.5000 = 0.5000$	0
$0.50000\times 2 = 1.0000 = 0$	1 ←LSB

乘法过程继续到小数为 0 或是所要的精度为止。然后，将"溢出"收集在一起，第一次溢出写在二进制小数点之后最高位上，一直写到最低位为止。数$0.11101_2 = 0.90625_{10}$。

如果十进制数包含整数和小数两部分，则必须将十进制小数点两边的整数和小数部分分开，分别完成相应的转换，然后，再把二进制整数和小数部分组合在一起。例如，将十进制数14.375_{10}转换成相应的二进制数：

$14.375 = 14_{10} + 0.375_{10}$

$14 \div 2 = 7$	余数	0 ←LSB
$7 \div 2 = 3$		1
$3 \div 2 = 1$		1
$1 \div 2 = 0$		1 ←MSB

$\boxed{14_{10} = 1110_2}$